COLE experiments at CLEF 2003 Spanish monolingual track

Jesús Vilares
Miguel A. Alonso
Departamento de Computación
Universidade da Coruña
Campus de Elviña s/n
15071 La Coruña (Spain)
{*jvilares,alonso*}*@udc.es*

Francisco J. Ribadas
Escuela Superior de Ingeniería Informática
Universidade de Vigo
Campus As Lagoas s/n
32004 Orense (Spain)
ribadas@uvigo.es

Abstract

In this our second participation in the CLEF Spanish monolingual track, we have continued applying Natural Language Processing techniques for single word and multi-word term conflation. Two different conflation approaches have been tested. The first approach is based on the lemmatization of the text in order to avoid inflectional variation. Our second approach consists of the employment of syntactic dependencies as complex index terms, in an attempt to solve the problems derived from syntactic variation and, in this way, to obtain more precise terms. Such dependencies are obtained through a shallow parser based on cascades of finite-state transducers.

1 Introduction

In Information Retrieval (IR) systems, the correct representation of a document through an accurate set of index terms is the basis for obtaining a good performance. If we are not able to both extract and weight appropriately the terms which capture the semantics of the text, this shortcoming will have an effect on all the subsequent processing.

In this context, one of the major limitations we have to deal with is the linguistic variation of natural languages [2], particularly when processing documents written in languages with more complex morphologic and syntactic structures than those present in English, as in the case of Spanish. When managing this type of phenomena, the employment of Natural Language Processing (NLP) techniques becomes feasible. This has been our working hypothesis since our research group, COLE Group[1], started its work on Spanish Information Retrieval. This was our working hypothesis in our participation in CLEF 2002 [18], and now in CLEF 2003.

As in our first participation, our main premise is the simplicity, motivated by the lack of available linguistic resources for Spanish such as large tagged corpora, treebanks or advanced lexicons. This work is a continuation and refinement of our previous work, presented in CLEF 2002, but centered this time on the employment of lemmatization for solving the *inflectional variation* and the employment of syntactic dependencies for solving the *syntactic variation*.

This article is outlined as follows. Section 2 describes the techniques used for single word term conflation. Our approach for dealing with syntactic variation through shallow parsing is introduced in Section 3. The tuning process of our system before the official runs is shown in Section 4. Finally, official runs are presented and discussed in Section 5.

[1] `http://www.grupocole.org`

2 Single word term conflation

As in our previous contribution to CLEF 2002 [18], our proposal for single word term conflation keeps being based on exploiting the lexical level in two phases: firstly, by solving the *inflectional variation* through lemmatization, and secondly, by solving the *derivational morphology* through the employment of morphological families.

The process followed for single word term conflation starts by tagging the document. The first step consists of applying our linguistically-motivated preprocessor module [9, 3] in order to perform tasks such as format conversion, tokenization, sentence segmentation, morphological pretagging, contraction splitting, separation of enclitic pronouns from verbal stems, expression identification, numeral identification and proper noun recognition. Classical approaches, such as stemming, rarely manage these phenomena, resulting in wrong simplifications during conflation process.

The output generated by our preprocessor is then taken as input by our tagger-lemmatizer, `MrTagoo` [6], although any high-performance part-of-speech tagger could be used instead. `MrTagoo` is based on a second order Hidden Markov Model (HMM), whose elements and procedures of estimation of parameters are based on Brant's work [4], and also incorporates certain capabilities which motivated its employment in our system. Such capabilities include a very efficient structure for storage and search —based on finite-state automata [8]—, management of unknown words, the possibility of integrating external dictionaries in the probabilistic frame defined by the HMM [10], and the possibility of management of segmentation ambiguity [7]

Nevertheless, these kind of tools are very sensitive to spelling errors, as, for example, in the case of sentences written completely in uppercase —e.g. news titles and subsection headings—, which cannot be correctly managed by the preprocessor and tagger modules. For this reason, the initial output of the tagger is processed by an *uppercase-to-lowercase* module [18] in order to process uppercase sentence, converting them to lowercase and restoring the spelling signs when necessary.

Once text has been tagged, the lemmas of the content words (nouns, verbs and adjectives) are extracted to be indexed. In this way we are solving the problems derived from inflection in Spanish. With regard to computational cost, the running cost of a lemmatizer-disambiguator is linear in relation to the length of the word, and cubic in relation to the size of the tagset, which is a constant. As we only need to know the grammatical category of the word, the tagset is small and therefore the increase in cost with respect to classical approaches (stemmers) becomes negligible.

Our previous experiments in CLEF 2002 showed that lemmatization performs better than stemming, even when using stemmers which also deals with derivational morphology.

Once inflectional variation has been solved, the next logical step consists on solving the problems caused by derivational morphology. For this purpose, we have grouped the words derivable one from another by means of mechanisms of derivational morphology; each one of these groups is a *morphological family*. Each one of the lemmas belonging to the same morphological family is conflated into the same term, a *representative* of the family. The set of morphological families are automatically generated from a large lexicon of Spanish words by means of a tool which implements the most common derivational mechanisms of Spanish [20]. Since the set of morphological families is generated statically, there is no increment in the running cost.

Nevertheless, our previous experiments in CLEF 2002 showed that the employment of morphological families for single word term conflation introduces too much noise in the system. This way, lemmatization will be the conflation technique to be used for single word term conflation, while morphological families will only be used in multi-word term conflation, as shown in Section 3.

3 Managing the syntactic variation through shallow parsing

Following the same scheme of our previous experiments, once we have established the way to process the content of the document at word level, the next step consists of deciding how to process, at phrase level, its syntactic content in order to manage the *syntactic variation* of the document. For this purpose, we will extract the pairs of words related through syntactic dependencies in order to use them as complex index terms. This process is performed in two steps: firstly, the text is parsed by means of a *shallow parser* and, secondly, the syntactic dependencies are extracted and conflated into index terms.

3.1 The shallow parser

When dealing with syntactic variation, we have to face the problems derived from the high computational cost of parsing. In order to maintain a linear complexity with respect to the length of the text to be analyzed, we have discarded the employment of full parsing techniques [14], opting for applying *shallow parsing* techniques, also looking for more robustness.

The theoretical basis for the design of our parser comes from formal language theory, which tells us that, given a context-free grammar and an input string, the syntactic trees of height k generated by a parser can be obtained by means of k layers of finite-state transducers: the first layer obtains the nodes labeled by non-terminals corresponding to left-hand sides of productions that only contain terminals on their right-hand side; the second layer obtains those nodes which only involve terminal symbols and those non-terminal symbols generated on the previous layer; and so on. It can be argued that the parsing capability of the system is, in this way, limited by the height of the parseable trees. Nevertheless, this kind of shallow parsing [1] has shown itself to be useful in several NLP application fields, particularly in Information Extraction. Its application in IR, which has not been deeply studied, has been tested by Xerox for English [11], showing its superiority with respect to classical approaches based on contiguous words.

This way, we have implemented a shallow parser based on a five layer architecture whose input is the output of our tagger-lemmatizer. Next, we will describe the function of each layer:

Layer 0: improving the preprocessing. Its function is the management of certain linguistic constructions in order to minimize the noise generated during the subsequent parsing. Such constructions include:

- *Numerals in non-numerical format.*
- *Quantity expressions.* Expressions of the type *algo más de dos millones* (a little more than two million) or *unas dos docenas* (about two dozens), which denote a number but with a certain vagueness about its concrete value, are identified as numeral phrases ($NumP$).
- *Expressions with a verbal function.* Some verbal expressions such as *tener en cuenta* (to take into account), must be considered as a unit, in this case synonym of the verb *considerar* (to consider), to avoid errors in the upper layers such as identifying *en cuenta* as a complement of the verb.

Layer 1: adverbial phrases and first level verbal groups. In this layer the system identifies, on the one hand, the *adverbial phrases* ($AdvP$) of the text, either those with an adverbial head —e.g. *rápidamente* (quickly)—, or those expressions not properly adverbial but with a equivalent function —e.g. *de forma rápida* (in a quick way)—. On the other hand, non-periphrastic verbal groups, which we name *first level verbal groups*, are processed, either their simple and compound forms, and either their active and passive forms.

Layer 2: adjectival phrases and second level verbal groups. Adjectival phrases ($AdjP$) such as *azul* (blue) or *muy alto* (very high) are managed here, together with periphrastic verbal groups, such as *tengo que ir* (I have to go), which we name *second level verbal groups*. *Verbal periphrasis* are unions of two or more verbal forms working as a unit, giving attributing shades of meaning, such as obligation, degree of development of the action, etc., to the semantics of the main verb. Moreover, these shades can not be expressed by means of the simple and compound forms of the verb.

Layer 3: noun phrases. In the case of noun phrases (NP), together with simple structures such as the attachment of determiners and adjectives to the name, we have considered more complex phenomena, such as the existence of *partitive complements* (PC) —e.g. *alguno de* (some of), *ninguno de* (none of)—, in order to cover more complex nominal structures —e.g. *cualquiera de aquellos coches nuevos* (any of those new cars)—.

Layer 4: prepositional phrases. Formed by a noun phrase (NP) preceded by a preposition (P), we have considered three different types according to this preposition, in order to make the extraction of dependencies easier: those preceded by the preposition *por* (by) or $PPby$, those preceded by *de* (of) or $PPof$, and the rest of prepositional phrases or PP.

Each of the rules involved in the different stages of the parsing process has been implemented through a finite-state transducer, compounding, in this way, a parser based on a cascade of finite-state transducers. Therefore, our approach maintains a linear complexity.

3.2 Extraction and conflation of dependencies

Once the text has been parsed, the system identifies the syntactic roles of the phrases recognized and extracts the following *dependency pairs*:

- A noun and each of its modifying adjectives.
- A noun and the head of its prepositional complement.
- The head of the subject and its predicative verb.
- The head of the subject and the head of the attribute. From a semantical point of view, copulative verbs are mere links, so the dependency is directly established between the subject and the attribute.
- An active verb and the head of its direct object.
- A passive verb and the head of its agent.
- A predicative verb and the head of its prepositional complement.
- The head of the subject and the head of a prepositional complement of the verb, but only when it is copulative (because of its special behavior).

Once such dependencies have been identified, they are conflated through the following conflation scheme:

1. The simple terms compounding the pair are conflated employing morphological families —see Section 2— in order to improve the management of the syntactic variation by covering the appearance of morphosyntactic variants of the original term [19, 12]. In this way, terms such as *cambio en el clima* (change of the climate) and *cambio climático* (climatic change), which express the same concept in different words —but semantically and derivatively related—, can be matched.
2. Conversion to lowercase and elimination of spelling signs, as in the case of stemmers. Previous experiments show that this process eliminates much of the noise introduced by spelling errors [18].

4 Tuning the system

Before making the official experiments for CLEF 2003, we tuned our parsing-based approach using the CLEF 2001/2002 corpus [13], formed by 215,738 news reports filling a total disk space of 509 MBs, and a set of 100 queries, from 41 to 140. The initial conditions of these training experiments were:

1. Employment of the vector-based indexing engine SMART [5], with an `atn-ntc` weighting scheme [17].
2. Stopword list obtained from the content word lemmas of the Spanish stopword list provided by SMART. In the case of the dependency pairs, a pair is eliminated if any of its compounding words is a stop word.
3. Employment of the uppercase-to-lowercase module to recover uppercase sentences during tagging.
4. Elimination of spelling signs and conversion to lowercase after conflation to reduce typographical errors.
5. The three fields of the query —*title*, *description* and *narrative*— were employed, but giving double relevance to the *title* statement because it summarizes the basic semantics of the query.

	lem	***sd1***	***sd2***	***sd3***	***sd4***	***sd5***	***sd6***	***sd7***	***sd8***	***sd9***	***sd10***	***sd11***	***sd12***	***opt***	**Δ**
Documents	99k	99k	99k	99k	99k	99k	99k	99k	99k	99k	99k	99k	99k	- -	- -
Relevant (5548 expected).	5220	5214	5250	**5252**	**5252**	5248	5249	5244	5242	5241	5240	5239	5239	5252	32
R-precision	.5131	.4806	.5041	.5137	.5175	.5174	.5200	**.5203**	.5197	.5182	.5175	.5158	.5167	.5203	.0072
Non-interpolated precision	.5380	.5085	.5368	.5440	.5461	.5462	.5464	**.5472**	.5463	.5462	.5462	.5459	.5456	.5472	.0092
Document precision	.5924	.5489	.5860	.5974	.6013	.6025	**.6028**	.6026	.6020	.6017	.6015	.6010	.6007	.6028	.0104
Precision at 0.00 Re.	.8754	.8493	.8729	.8716	.8689	.8684	.8686	.8706	.8681	.8654	.8696	.8735	**.8760**	.8760	.0006
Precision at 0.10 Re.	.7934	.7602	.8027	.8019	.8079	.8093	.8082	.8069	.8071	.8090	**.8097**	.8088	.8076	.8097	.0163
Precision at 0.20 Re.	.7340	.6847	.7240	.7394	.7435	.7440	.7465	**.7468**	.7458	.7456	.7433	.7410	.7387	.7468	.0128
Precision at 0.30 Re.	.6697	.6355	.6671	.6777	**.6835**	.6802	.6826	.6819	**.6835**	.6825	.6823	.6823	.6820	.6835	.0138
Precision at 0.40 Re.	.6256	.5911	.6206	.6297	.6322	.6332	**.6348**	.6335	.6324	.6324	.6323	.6320	.6322	.6348	.0092
Precision at 0.50 Re.	.5749	.5384	.5707	.5825	**.5856**	.5827	.5843	.5849	.5842	.5840	.5830	.5831	.5825	.5856	.0107
Precision at 0.60 Re.	.5146	.4753	.5041	.5137	.5168	.5176	.5187	**.5214**	.5207	.5208	.5195	.5198	.5195	.5214	.0068
Precision at 0.70 Re.	.4402	.4142	.4331	.4408	.4428	.4430	.4445	.4462	**.4467**	.4457	.4454	.4450	.4451	.4467	.0065
Precision at 0.80 Re.	.3652	.3512	.3691	.3714	.3724	.3724	.3722	**.3738**	.3733	.3729	.3728	.3723	.3727	.3738	.0086
Precision at 0.90 Re.	.2723	.2649	.2799	.2834	**.2853**	.2850	.2830	.2831	.2817	.2813	.2808	.2805	.2792	.2853	.0130
Precision at 1.00 Re.	.1619	.1534	.1613	.1645	.1628	.1634	.1630	**.1646**	.1641	.1641	.1641	.1641	.1638	.1646	.0027
Precision at 5 docs.	.6747	.6525	**.6909**	.6869	.6848	.6788	.6808	.6828	.6808	.6828	.6828	.6808	.6747	.6909	.0162
Precision at 10 docs.	.6010	.5859	.6091	.6192	**.6202**	.6192	.6192	.6172	.6152	.6121	.6131	.6121	.6131	.6202	.0192
Precision at 15 docs.	.5623	.5441	.5690	.5737	.5778	**.5791**	**.5791**	.5764	.5758	.5758	.5737	.5731	.5710	.5791	.0168
Precision at 20 docs.	.5374	.5040	.5298	.5328	.5354	.5343	.5384	.5394	.5384	**.5399**	**.5399**	**.5399**	**.5399**	.5399	.0025
Precision at 30 docs.	.4825	.4549	.4778	.4852	.4892	.4886	.4882	**.4896**	**.4896**	.4892	.4886	.4879	.4875	.4896	.0071
Precision at 100 docs.	.3067	.2873	.3017	.3070	.3084	**.3095**	.3087	.3089	.3083	.3083	.3084	.3088	.3088	.3095	.0028
Precision at 200 docs.	.2051	.1959	.2033	.2057	.2062	.2063	**.2067**	**.2067**	.2065	.2062	.2063	.2063	.2064	.2067	.0016
Precision at 500 docs.	.0997	.0980	.0997	.1001	.1004	**.1005**	**.1005**	**.1005**	**.1005**	.1004	.1004	.1004	.1003	.1005	.0008
Precision at 1000 docs.	.0527	.0527	.0530	**.0531**	**.0531**	.0530	.0530	.0530	.0529	.0529	.0529	.0529	.0529	.0531	.0004

Table 1: Tuning the system with the CLEF 2001/2002 corpus

6. Combination of simple and complex terms. The former, obtained through the lemmatization of the content words of the text, the later, obtained through the conflation of the syntactic dependencies identified by the shallow parser.

For this training phase we used the indexing of the content words lemmas of the text (*lem*) as as our point of reference, since our previous experiments [18, 19], where lemmatization beats stemming as word-level conflation technique, indicate that this technique is the best starting point for the development of NLP-based conflation methods.

Table 1 shows the results obtained. The first column of the table (*lem*) shows the results for lemmatization, whereas the next columns (*sdx*) contain the results obtained by merging lemmatized simple terms and complex terms based on syntactic dependencies (*sd*), when the weight relation between simple and complex terms, x to 1, changes. The column *opt* is formed by the best results obtained with *sd* for each parameter considered, which are also highlighted in bold. Finally, the column Δ shows the improvement of *opt* with respect to *lem*. Each row contains one of the parameters employed to measure the performance of the system: number of documents retrieved, number of relevant documents retrieved (5548 expected), R-precision, average precision (non-interpolated) for all relevant documents (averaged over queries), average document precision for all relevant documents (averaged over relevant documents), precision at standard levels of recall, and precision at N documents retrieved.

As is shown in column *sd1*, the direct employment of syntactic dependencies as index terms led to a general decrease of the performance of the system. After examining the behavior of the system for each query, we inferred that the problem was caused by an over-balance of the weight of complex terms, which are much less frequent than simple terms and, therefore, with a much higher assigned weight. In this way, when a matching between a complex term and a relevant document occurred, its assigned score increased substantially, improving its ranking. Nevertheless, in the same way, when a undesired matching with a non-relevant document occurred, its computed relevance grew excessively. It can be argued that, according to this, we would expect similar results to those obtained only with simple terms. Nevertheless, it should be noticed that complex term matchings are much less frequent than those for simple terms. Therefore, incorrect matchings between complex terms and non-relevant documents are much more harmful than those for simple terms, whose effect tends to be weakened by the rest of the matchings. It can be deduced that this first attempt led to a increasing instability of the system.

In order to minimize the negative effect of undesired matchings, the over-balance of complex terms needed to be solved. Therefore, the balance factor between the weights of simple and complex terms was corrected, decreasing the extra initial relevance assigned to complex terms. The results of this solution were immediate, as is shown in the remaining *sdx* columns, where the performance of the system gradually improves, particularly with respect to the precision in the first 15 documents retrieved and to the number of relevant documents retrieved (5220 with *lem*, 5214 with *sd1*, and 5250 with *sd2*).

As generally happens in IR, we can not talk about a best method for all situations. From a ranking point of view and with respect to the top N documents retrieved, *sd4* —in which the weights of simple terms are quadrupled— obtained the best results, also reaching the best recall (5252 relevant documents retrieved). Nevertheless, the best results for global performance measures were obtained with *sd7*, using a higher balance factor. The performance of the system gets worse, in general, for higher factors, except in the case of the precision vs. recall, where we obtain the best results for the lowest levels of recall, nevertheless, at the expense of sacrificing performance in the rest of aspects.

Since our priority was to increase the precision of the top documents retrieved, we decided to use a balance factor of 4, as in the case of *sd4*, for the official runs.

5 CLEF 2003 official runs

In this new edition of CLEF, the document corpus for the Spanish Monolingual Track has been enlarged. The new corpus is formed by 215,738 news (509 MB) from 1994 plus 238,307 news (577 MB) from 1995; that is, 454,045 documents (1086 MB). The set of topics has also been enlarged; this year it consists of 60 queries (141 to 200) instead of 50 as previous years.

Our group submitted four runs to the CLEF 2003 Spanish monolingual track:

- `coleTDlemZP03` (`TDlemZP` for short): Conflation of content words via lemmatization, i.e. each form of a content word is replaced by its lemma. This kind of conflation takes only into account inflectional morphology. The resulting conflated document was indexed using the probabilistic engine ZPrise[2], employing the Okapi BM25 weight scheme [15] with the constants defined in [16] for Spanish ($b = 0.5$, $k_1 = 2$). The query is formed by the set of meaning lemmas present in the *title* and *description* fields.

- `coleTDNlemZP03` (`TDNlemZP` for short): The same as before, but the query also includes the set of meaning lemmas obtained from the *narrative* field.

- `coleTDNlemSM03` (`TDNlemSM` for short): As in the case of `coleTDNlemZP03`, the three fields of the query are conflated through lemmatization. Nevertheless, this time the indexing engine is the vector-based SMART [5], with an `atn-ntc` weighting scheme [17]. This run was submitted in order to use it as a point of reference for the rest of runs.

- `coleTDNpdsSM03` (`TDNpdsSM` for short): Text conflated via the combination of simple terms, obtained through the lemmatization of content words, and complex terms, obtained through the conflation of syntactic dependencies, as was described in Section 3. The balance factor between the weights of simple and complex terms is 4 to 1 —i.e. the weights of simple terms are quadrupled— looking for increasing the precision of the top ranked documents according to the results of Section 4.

There is no experiments indexing syntactic dependencies with the Okapi BM25 weight scheme, since we are still studying the best way to integrate them into a probabilistic model. With respect to the conditions employed in the official runs, they were:

1. Stopword list obtained from the content word lemmas of the SMART Spanish stopword list.

2. Employment of the uppercase-to-lowercase module to recover uppercase sentences during tagging.

[2] `http://www.itl.nist.gov`

Table 2: CLEF 2003: performance measures

	TDlemZP	TDNlemZP	TDNlemSM	TDNpdsSM
Documents retrieved	57,000	57,000	57,000	57,000
Relevant documents retrieved (2368 expected)	2,237	2,253	2,221	2,249
R-precision	0.4503	0.4935	0.4453	0.4684
Average non-interpolated precision	0.4662	0.5225	0.4684	0.4698
Average document precision	0.5497	0.5829	0.5438	0.5408
11-points average precision	0.4788	0.5325	0.4799	0.4861

Table 3: CLEF 2003: average precision at 11 standard recall levels and at N seen documents

Recall	Precision				N	Precision			
	TDlemZP	TDNlemZP	TDNlemSM	TDNpdsSM		TDlemZP	TDNlemZP	TDNlemSM	TDNpdsSM
0.00	0.8014	0.8614	0.7790	0.7897	5	0.5930	0.6421	0.5930	0.5684
0.10	0.7063	0.7905	0.6982	0.7165	10	0.5070	0.5596	0.5018	0.4965
0.20	0.6553	0.7301	0.6331	0.6570	15	0.4713	0.4971	0.4515	0.4573
0.30	0.5969	0.6449	0.5738	0.6044	20	0.4307	0.4614	0.4281	0.4202
0.40	0.5485	0.5911	0.5388	0.5562	30	0.3719	0.4012	0.3784	0.3678
0.50	0.4969	0.5616	0.5003	0.5092	100	0.2316	0.2393	0.2316	0.2305
0.60	0.4544	0.4871	0.4457	0.4391	200	0.1461	0.1505	0.1455	0.1458
0.70	0.3781	0.4195	0.3987	0.3780	500	0.0726	0.0731	0.0718	0.0719
0.80	0.3083	0.3609	0.3352	0.3191	1000	0.0392	0.0395	0.0390	0.0395
0.90	0.2093	0.2594	0.2292	0.2248					
1.00	0.1111	0.1512	0.1472	0.1525					

3. Elimination of spelling signs and conversion to lowercase after conflation to reduce typographical errors.

4. Except for the first run, TDlemZP, the terms extracted from *title* field of the query are given double relevance with respect to *description* and *narrative*.

According to Tables 2 and 3, the probabilistic-based approach through a BM25 weighting scheme —TDlemZP and TDNlemZP— shows clearly superior to the vector-based atn-ntc weighting scheme —TDNlemSM and TDNpdsSM—, even when only lemmatizing the text. As we can see, TDlemZP obtains similar or better results than TDNlemSM even when the later also employs the extra information provided by the *narrative* field of the topic.

With respect to the main contribution of this work, the employment of syntactic dependencies as complex index terms, the results are a little different from expected. With respect to global performance measures, TDNpdsSM run obtains better results than TDNlemSM, except for average document precision. Nevertheless, the behavior of the system with respect to ranking has changed partially, since the results obtained for precision at N documents retrieved when employing complex terms —TDNpdsSM— are worse than those obtained using only simple lemmatized terms —TDNlemSM—. On the other hand, the results for precision vs. recall keep being better.

Taking into account the possibility that the weight balance factor of 4 employed in the official run was not the most accurate for these set of queries, we have tried different values in a range of 1 to 12, as is shown in Table 4. The scheme of these extra experiments is the same followed during the training phase —see Table 1—. The first column, *lem*, shows the results for lemmatization —i.e. TDNlemSM— whereas *sdx* columns contain the results obtained using syntactic dependencies with a weight balance factor of x.You are reminded that *sd4* shows the results for the official run TDNpdsSM, because it was created using a balance factor of $x = 4$. The column *opt* shows the best results obtained for *sdx* and the column Δ shows the improvement of *opt* with respect to *lem*.

The results obtained make even more difficult to choose a balance factor, since the degree of improvement with respect to lemmatization changes according to the balance factor. It could be considered that the best balance factor for global measures is 10 —*sd10*—, since it obtains the best non-interpolated and document precision, very

	lem	***sd1***	***sd2***	***sd3***	***sd4***	***sd5***	***sd6***	***sd7***	***sd8***	***sd9***	***sd10***	***sd11***	***sd12***	***opt***	Δ
Documents	57k	57k	57k	57k	*57k*	57k	57k	57k	57k	57k	57k	57k	57k	- -	- -
Relevant (2368 expected).	2221	2218	2241	2243	***2249***	2244	2244	2245	2244	2243	2243	2242	2240	2249	28
R-precision	.4453	.4121	.4581	.4637	***.4684***	.4540	.4503	.4490	.4491	.4487	.4493	.4502	.4496	.4684	.0031
Non-interpolated precision	.4684	.4132	.4481	.4627	*.4698*	.4664	.4683	.4689	.4719	**.4723**	**.4723**	.4714	.4717	.4723	.0039
Document precision	.5438	.4664	.5163	.5329	*.5408*	.5438	.5456	.5471	.5481	.5483	**.5485**	.5481	.5484	.5485	.0047
Precision at 0.00 Re.	.7790	.7926	**.8151**	.8049	*.7897*	.7819	.7822	.7806	.7852	.7817	.7820	.7814	.7898	.8151	.0361
Precision at 0.10 Re.	.6982	.6645	.6873	**.7185**	*.7165*	.7027	.6999	.7030	.7141	.7127	.7127	.7123	.7161	.7185	.0203
Precision at 0.20 Re.	.6331	.5803	.6237	.6365	***.6570***	.6542	.6521	.6509	.6473	.6457	.6431	.6390	.6374	.6570	.0239
Precision at 0.30 Re.	.5738	.5288	.5711	.5919	***.6044***	.5903	.5934	.5928	.5924	.5913	.5904	.5863	.5865	.6044	.0306
Precision at 0.40 Re.	.5388	.4864	.5309	.5495	***.5562***	.5442	.5460	.5459	.5431	.5448	.5444	.5428	.5417	.5562	.0174
Precision at 0.50 Re.	.5003	.4376	.4770	.5008	***.5092***	.4966	.5013	.4992	.5070	.5081	.5088	.5072	.5067	.5092	.0089
Precision at 0.60 Re.	.4457	.3698	.4194	.4308	*.4391*	.4382	.4416	.4416	.4479	**.4498**	.4476	.4477	.4469	.4498	.0041
Precision at 0.70 Re.	.3987	.3137	.3497	.3665	*.3780*	.3844	.3898	.3921	.3947	.3959	**.3972**	.3970	.3966	.3972	-.0015
Precision at 0.80 Re.	.3352	.2597	.2973	.3092	*.3191*	.3248	.3288	.3294	.3304	.3312	.3320	**.3321**	.3320	.3321	-.0031
Precision at 0.90 Re.	.2292	.1950	.2168	.2208	*.2248*	.2268	.2301	.2316	.2328	**.2330**	.2307	.2308	.2310	.2330	.0008
Precision at 1.00 Re.	.1472	.1317	.1508	**.1534**	*.1525*	.1501	.1499	.1499	.1505	.1506	.1491	.1490	.1491	.1534	.0062
Precision at 5 docs.	.5930	.4947	.5333	.5684	*.5684*	.5719	.5789	.5860	.5930	**.6000**	**.6000**	.5930	.5895	.6000	.0070
Precision at 10 docs.	.5018	.4421	.4877	.4912	*.4965*	.5035	.5053	**.5105**	.5070	.5070	**.5105**	.5088	.5088	.5105	.0087
Precision at 15 docs.	.4515	.3895	.4421	.4503	***.4573***	.4526	.4538	.4538	.4538	.4526	.4526	.4503	.4526	.4573	.0018
Precision at 20 docs.	.4281	.3640	.4009	.4140	*.4202*	.4211	.4211	.4219	.4237	.4237	.4246	.4246	**.4254**	.4254	.0027
Precision at 30 docs.	.3784	.3234	.3509	.3667	*.3678*	.3737	.3754	.3772	.3778	.3807	.3789	.3813	**.3819**	.3819	.0035
Precision at 100 docs.	.2316	.2053	.2221	.2289	*.2305*	.2330	.2335	**.2342**	.2340	.2337	.2333	.2332	.2333	.2342	.0026
Precision at 200 docs.	.1455	.1367	.1430	.1448	*.1458*	.1459	**.1466**	.1465	**.1466**	.1464	.1463	.1463	.1463	.1466	.0011
Precision at 500 docs.	.0718	.0691	.0712	.0716	*.0719*	.0719	.0720	**.0721**	**.0721**	**.0721**	**.0721**	**.0721**	.0720	.0721	.0003
Precision at 1000 docs.	.0390	.0389	.0393	.0394	***.0395***	.0394	.0394	.0394	.0394	.0394	.0394	.0393	.0393	.0395	.0005

Table 4: CLEF 2003: Re-tuning the system a posteriori

good recall —2243 documents retrieved, against 2221 for *lem* and 2249 for *opt*—, and a slight improvement for R-precision.

From a ranking point of view, our official run, *sd4* —i.e. `TDNpdsSM`—, is the best compromise when talking about precision vs. recall, since it obtains the best results in the range 0.20–0.60, and very good results for the range 0.00–0.20. Nevertheless, its results for precision at N documents retrieved are not very good, since there is no improvement with respect to *lem*, which was our goal. In this case, *sd10* shows again as the best option, since it obtains the best compromise for the top 15 documents retrieved; however, the improvement reached is lesser than the one obtained during the training phase —see Table 1—.

Acknowledgements

The research described in this paper has been supported in part by Ministerio de Ciencia y Tecnología (TIC2000-0370-C02-01, HP2001-0044 and HF2002-81), FPU grants of Secretaría de Estado de Educación y Universidades, Xunta de Galicia (PGIDT01PXI10506PN, PGIDIT02PXIB30501PR and PGIDIT02SIN01E) and Universidade da Coruña. The authors also would like to thank Darrin Dimmick, from NIST, for giving us the opportunity to use the ZPrise system, and Fernando Martínez, from Universidad de Jaén, for helping us to make it operative.

References

[1] S. Abney. Partial parsing via finite-state cascades. *Natural Language Engineering*, 2(4):337–344, 1997.

[2] A. Arampatzis, T. van der Weide, C. Koster, and P. van Bommel. Linguistically motivated information retrieval. In *Encyclopedia of Library and Information Science*. Marcel Dekker, Inc., New York and Basel, 2000.

[3] Fco. Mario Barcala, Jesús Vilares, Miguel A. Alonso, Jorge Graña, and Manuel Vilares. Tokenization and proper noun recognition for information retrieval. In A Min Tjoa and Roland R. Wagner, editors, *Thirteen In-*

ternational Workshop on Database and Expert Systems Applications. 2-6 September 2002. Aix-en-Provence, France, pages 246–250, Los Alamitos, California, USA, September 2002. IEEE Computer Society Press.

[4] Thorsten Brants. TNT - a statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP'2000)*, Seattle, 2000.

[5] C. Buckley. Implementation of the SMART information retrieval system. Technical report, Department of Computer Science, Cornell University, 1985. Source code available at `ftp://ftp.cs.cornell.edu/pub/smart`.

[6] Jorge Graña. *Técnicas de Análisis Sintáctico Robusto para la Etiquetación del Lenguaje Natural*. PhD thesis, University of La Coruña, La Coruña, Spain, 2000.

[7] Jorge Graña, Miguel A. Alonso, and Manuel Vilares. A common solution for tokenization and part-of-speech tagging: One-pass Viterbi algorithm vs. iterative approaches. In Petr Sojka, Ivan Kopecek, and Karel Pala, editors, *Text, Speech and Dialogue*, volume 2448 of *Lecture Notes in Computer Science*, pages 3–10. Springer-Verlag, Berlin-Heidelberg-New York, 2002.

[8] Jorge Graña, Fco. Mario Barcala, and Miguel A. Alonso. Compilation methods of minimal acyclic automata for large dictionaries. In Bruce W. Watson and Derick Wood, editors, *Implementation and Application of Automata*, volume 2494 of *Lecture Notes in Computer Science*, pages 135–148. Springer-Verlag, Berlin-Heidelberg-New York, 2002.

[9] Jorge Graña, Fco. Mario Barcala, and Jesús Vilares. Formal methods of tokenization for part-of-speech tagging. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 2276 of *Lecture Notes in Computer Science*, pages 240–249. Springer-Verlag, Berlin-Heidelberg-New York, 2002.

[10] Jorge Graña, Jean-Cédric Chappelier, and Manuel Vilares. Integrating external dictionaries into stochastic part-of-speech taggers. In *Proceedings of the Euroconference Recent Advances in Natural Language Processing (RANLP 2001)*, pages 122–128, Tzigov Chark, Bulgaria, 2001.

[11] D. A. Hull, G. Grefenstette, B. M. Schulze, E. Gaussier, H. Schutze, and J. O. Pedersen. Xerox TREC-5 site report: routing, filtering, NLP, and Spanish tracks. In *Proceedings of the Fifth Text REtrieval Conference (TREC-5)*, pages 167–180, 1997.

[12] Christian Jacquemin and Evelyne Tzoukermann. NLP for term variant extraction: synergy between morphology, lexicon and syntax. In Tomek Strzalkowski, editor, *Natural Language Information Retrieval*, volume 7 of *Text, Speech and Language Technology*, pages 25–74. Kluwer Academic Publishers, Dordrecht/Boston/London, 1999.

[13] C. Peters, editor. *Results of the CLEF 2002 Cross-Language System Evaluation Campaign, Working Notes for the CLEF 2002 Workshop*, Rome, Italy, Sept. 2002. Official site of CLEF: `http://www.clef-campaign.org`

[14] Jose Perez-Carballo and Tomek Strzalkowski. Natural language information retrieval: progress report. *Information Processing and Management*, 36(1):155–178, 2000.

[15] Okapi/Keenbow at TREC-8. In E. Voorhees and D. K. Harman, editors, *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, NIST Special Publication 500-264, pages 151–161, 2000.

[16] J. Savoy. Report on CLEF-2002 Experiments: Combining Multiple Sources of Evidence. In [13], pages 31–46.

[17] J. Savoy, A. Le Calve, and D. Vrajitoru. Report on the TREC-5 experiment: Data fusion and collection fusion. Proceedings of TREC'5, NIST publication #500-238, pages 489–502, Gaithersburg, MD, 1997.

[18] Jesús Vilares, Miguel A. Alonso, Francisco J. Ribadas, and Manuel Vilares. COLE experiments at CLEF 2002 Spanish monolingual track. In C. Peters, M. Braschler, J. Gonzalo, and M. Kluck, editors, *Advances in Cross-Language Information Retrieval: Results of the CLEF 2002 Evaluation Campaign*, volume 2785 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin-Heidelberg-New York, 2003.

[19] Jesús Vilares, Fco. Mario Barcala, and Miguel A. Alonso. Using syntactic dependency-pairs conflation to improve retrieval performance in Spanish. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 2276 of *Lecture Notes in Computer Science,*, pages 381–390. Springer-Verlag, Berlin-Heidelberg-New York, 2002.

[20] Jesús Vilares, David Cabrero, and Miguel A. Alonso. Applying productive derivational morphology to term indexing of Spanish texts. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 2004 of *Lecture Notes in Computer Science*, pages 336–348. Springer-Verlag, Berlin-Heidelberg-New York, 2001.